02

Otros tipos de Sets y Iterators

Transcripción

[00:00] Hola, hola. Vamos a iniciar nuestra clase 16. Nuestra clase 16 está dentro del bloque otros tipos de Sets e Iterators. Para esto vamos a duplicar nuestra clase 15, vamos a colocar clase 16, vamos a remover esto de aquí y hagamos lo siguiente. Vamos a recorrer primero esta lista. Sería curso1.getAlumnos. Vamos a recorrer nuestra lista de alumnos .forEach, ahora ponemos aquí alumno.

[00:40] En lambda, por ejemplo aquí, si deseo puedo utilizar las llaves, si deseo no, porque solamente voy a colocar una sola línea. Es tipo como si fuese un if o un else. No es necesario si yo coloco solamente una línea, System.out.println, coloco aquí alumno. Una vez hecho esto, ¡voilá! Imprimió nuestra lista de alumnos.

[01:08] Ahora supongamos que nosotros en nuestra clase curso queramos los alumnos de acuerdo a como yo los adicioné. Vamos a utilizar el LinkedHashSet. Una vez hecho esto, ejecutamos y tenemos ahí Luis Miguel, Pepito los Palotes, Juan Carlos, Pedro Pedrito, todo en orden. No quiere decir que ahora yo puedo seleccionar un orden o puedo seleccionar una posición. No.

[01:45] Quiere decir que él adicionó en orden y que cuando yo haga el recorrido, va a aparecer en orden. No confundirnos por esa parte. Que el get viene más del lado de las listas, de los list, del interface list. Una vez hecho esto, vamos a dejarlo como un pequeño backup. Vamos a dejarlo aquí en hashSet. ¿Qué cosa es un iterator?

[02:19] Un iterator es una forma en cómo nosotros vamos a leer una lista, un collection, por ejemplo. Aquí utilizamos Java.util iterator. Aquí colocamos ¿qué cosa vamos a recorrer? Alumno. Una vez que hacemos esto, colocamos el nombre de nuestra variable alumnoIterator, colocamos aquí new. New no. ¿Por qué? Porque ya tenemos la lista, entonces tenemos la lista curso1.getAlumnos().iterator.

[02:56] ¿Ahí que me está devolviendo? Me está devolviendo prácticamente unos iteradores para poder recorrer nuestra lista de getAlumnos. Pero yendo aquí a nuestro gráfico, por ejemplo, ¿qué va a hacer un iterator? Un iterator va a ir primero a recorrer aquí, después el iterator pasa aquí, después pasa aquí, después pasa aqui y despues pasa aquí. Ya llegó al final, acabó.

[03:27] Si quiero hacer una iteración más, va a dar un error. Si quiero ir para atrás, no puedo. Tendría que hacer de nuevo el iterator nuevamente, para que él comience de nuevo de cero. Entonces haciendo esto sería así: alumnoIterator.hashNext. Ahí él quiere decir, va a ir uno por uno hasta llegar al final. Por ejemplo aquí vamos a utilizar así un while.

[04:02] ¿Por qué me devuelve un true o false? Entonces cuando es true quiere decir que tenemos el valor, y aquí está. Y aquí hacemos un System.out.println colocamos aquí alumnoIterator.next. Estamos diciendo "dame el alumno 7", el alumno que yo quiero. Dame por ejemplo el primer alumno, después el segundo, después el tercero. Y con el otro estoy validando simplemente si existe un siguiente.

[04:54] Aquí vamos a imprimir. Para no confundir, vamos aquí a comentar este forEach, vamos a ver aquí. Imprimió nuestra misma lista. Perfecto. Ahora supongamos que yo quiero aquí, hacer alumnoIterator.next. En teoría aquí yo ya recorrí, o sea, quiere decir que ya llegué aquí al final.

[05:27] El next quiere decir que no va a haber nada. ¿que va a hacer aquí? ¿Qué va a suceder? Vamos a verlo aquí. Error. ¿Por qué? Porque no existe elemento,

no encontró un elemento. Entonces esto sería un poco, vimos ahora nuestra clase, vimos un Linked HashSet, vimos cómo recorrer también otra lista utilizando iterator y esto sería todo en nuestra clase 16. Muchas gracias.